AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

Claim 1 (Currently amended): A system for performing kernel-mode operations comprising:

a kernel-mode interface generator for dynamically generating a kernel-mode interface driver, which in turn generates a call gate to perform a kernel-mode operation with kernel-mode authorization in a kernel mode; and

an authorization interface, coupled to the kernel-mode interface generator, to connect a user mode to kernel mode, switching a process from user mode to kernel mode via the call gate to perform the kernel-mode operation.

Claim 2 (Original): The system as claimed in claim 1, wherein the authorization interface sends a call gate request to the kernel-mode interface generator to generate the kernel-mode interface driver, the call gate generated accordingly, and the authorization interface instructing the process to enter the kernel mode through the call gate.

Claim 3 (Original): The system as claimed in claim 1, wherein the kernel-mode operation is an operation with Ring 0 authorization level in the kernel mode.

Claim 4 (Original): The system as claimed in claim 1, wherein the process, a user-mode operation, is capable of user-mode authorization in a protected mode.

Claim 5 (Original): The system as claimed in claim 4, wherein the user-mode authorization is Ring 3 authorization level in the protected mode.

Claim 6 (Original): The system as claimed in claim 4, wherein the call gate sets a call gate selector and an entry point in a global descriptor table, having a call gate descriptor and a code-segment descriptor, to enable the process to perform the operation with kernel-mode authorization in the kernel mode.

Claim 7 (Original): The system as claimed in claim 6, wherein the user-mode authorization of the process is switched to kernel-mode authorization by the call gate selector via the entry point in the global descriptor table, and is switched back after the operation with kernel-mode authorization has been performed.

Claim 8 (Original): The system as claimed in claim 7, wherein a far call stated by the call gate selector points to the call gate descriptor, and a CPU switches an instruction pointer to the entry point, when a caller from the call gate gives a call, if the caller has kernel-mode authorization.

Claim 9 (Original): The system as claimed in claim 8, wherein the instruction pointer has kernel-mode authorization, is switched to the entry point, to perform the operation with kernel-mode authorization in the kernel mode, and is switched back to the user-mode authorization after the operation with kernel-mode authorization has been performed.

Claim 10 (Currently amended): A method for performing kernel-mode operations comprising steps of:

providing a kernel-mode interface generator;

dynamically generating a kernel-mode interface driver using the kernel-mode interface generator ~~to generate~~ ;

generating a call gate performing a kernel-mode operation with kernel-mode authorization in a kernel mode using the kernel-mode interface driver;

providing an authorization interface to connect a user mode to the kernel mode; and

switching a process from the user mode to the kernel mode via the call gate through the authorization interface to perform the kernel-mode operation with kernel-mode authorization.

Claim 11 (Original): The method as claimed in claim 10, further comprising, in the step of providing the authorization interface:

sending of a call gate request by the authorization interface to the kernel-mode interface generator to generate the kernel-mode interface driver;

generating the call gate using the kernel-mode interface driver; and

the authorization interface instructing the process to enter the kernel mode through the call gate.

Claim 12 (Original): The method as claimed in claim 10, wherein in the generating step, the kernel-mode operation is an operation with Ring 0 authorization level in the kernel mode.

Claim 13 (Original): The method as claimed in claim 10, wherein in the switching step, the process, a user-mode operation, is capable of user-mode authorization in a protected mode.

Claim 14 (Original): The method as claimed in claim 13, wherein the user-mode authorization is Ring 3 authorization level in the protected mode.

Claim 15 (Original): The method as claimed in claim 13, wherein in the generating step, the call gate sets a call gate selector and an entry point in a global descriptor table, having a call gate descriptor and a code-segment descriptor, to perform the operation with kernel-mode authorization in the kernel mode.

Claim 16 (Currently amended): The method as claimed in claim [[14]] 15, wherein the user-mode authorization of the process is switched to kernel-mode authorization by the call gate selector via the entry point in the global descriptor table, and is switched back after the operation with kernel-mode authorization has been performed.

Claim 17 (Original): The method as claimed in claim 16, wherein a far call stated by the call gate selector points to the call gate descriptor, a CPU switches an instruction pointer to the entry point, when a caller from the call gate sends a call, if the caller has kernel-mode authorization.

Claim 18 (Original): The method as claimed in claim 17, wherein the instruction pointer has kernel-mode authorization, is switched to the entry point, to perform the operation with kernel-mode authorization in the kernel mode, and is switched back to user-mode authorization after performing the operation with kernel-mode authorization.

Claim 19 (Currently amended): A storage medium for storing a computer program providing a method for performing kernel-mode operations, comprising using a computer to perform the steps of:

providing a kernel-mode interface generator;

dynamically generating a kernel-mode interface driver using the kernel-mode interface generator ~~to generate~~ ;

generating a call gate ~~with~~ performing a kernel-mode operation with kernel-mode authorization in a kernel mode using the kernel-mode interface driver;

providing an authorization interface to connect a user mode to the kernel mode; and

switching a process from the user mode to the kernel mode via the call gate through the authorization interface to perform the kernel-mode operation with kernel-mode authorization.

Claim 20 (Original): The storage medium as claimed in claim 19, wherein the authorization interface sends a call gate request to the kernel-mode interface generator to generate the kernel-mode interface driver, the call gate is generated according thereto, and the authorization interface directs the process to enter the kernel mode through the call gate.

Claim 21 (Original): The storage medium as claimed in claim 19, wherein the kernel-mode operation is an operation with Ring 0 authorization level in the kernel mode.

Claim 22 (Original): The storage medium as claimed in claim 19, wherein the process, a user-mode operation, is capable of user-mode authorization in a protected mode.

Claim 23 (Original): The storage medium as claimed in claim 22, wherein the user-mode authorization is Ring 3 authorization level in the protected mode.

Claim 24 (Original): The storage medium as claimed in claim 23, wherein the call gate sets a call gate selector and an entry point in a global descriptor table, having a call gate descriptor and a code-segment descriptor, to perform the operation with kernel-mode authorization in the kernel mode.

Claim 25 (Original): The storage medium as claimed in claim 24, wherein the user-mode authorization of the process is switched to kernel-mode authorization by the call gate selector via the entry point in the global descriptor table, and is switched back after the operation with kernel-mode authorization has been performed.

Claim 26 (Original): The storage medium as claimed in claim 25, wherein a far call stated by the call gate selector points to the call gate descriptor, a CPU switches an instruction pointer to the entry point, when a caller from the call gate sends a call, if the caller has kernel-mode authorization.

Claim 27 (Original): The storage medium as claimed in claim 26, wherein the instruction pointer has kernel-mode authorization, is switched to the entry point, to perform the operation with kernel-mode authorization in the kernel mode, and is switched back to user-mode authorization after performing the operation with kernel-mode authorization.